

06-22/00 1c816 U.S. PTO 09/602576 06/22/00

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF EXPRESS MAILING

I hereby certify that this paper and the documents and/or fees referred to as attached therein are being deposited with the United States Postal Service on June 22, 2000 in an envelope as "Express Mail Post Office to Addressee" service under 37 CFR §1.10, Mailing Label Number EL560312993US, addressed to Box Patent Application, Assistant Commissioner for Patents, Washington, DC 20231.

Attorney Docket No.: BIG1P001

First Named Inventor: Joshua A. Jacobs

Dionna Holmes

Dionna Holmes

UTILITY PATENT APPLICATION TRANSMITTAL (37 CFR. § 1.53(b))
(Regular application claiming priority of a provisional)

Assistant Commissioner for Patents
Box Patent Application
Washington, DC 20231

☐ Duplicate for
fee processing

Sir: This is a request for filing a patent application under 37 CFR. § 1.53(b) in the name of inventors:
**Joshua A. Jacobs, John F. Shiple, Christopher R. Miller, Mahesh Tyagarajan,
Kirt David Johnson, Bernard Luke Gallagher, Chris Kiernan**

For: **AN INTEGRATED PLATFORM FOR DEVELOPING AND MAINTAINING A
DISTRIBUTED MULTIAPPLICATION ONLINE PRESENCE**

Priority of U.S. Provisional Application No. 60/142,181 filed on July 1, 1999 is claimed under
35 U.S.C. § 119(e).

Application Elements:

- ☒ 40 Pages of Specification, Claims and Abstract
☒ 14 Sheets of **informal** Drawings
☒ 3 Pages Combined Declaration and Power of Attorney

Accompanying Application Parts:

- ☒ Assignment and Assignment Recordation Cover Sheet (recording fee of \$40.00 enclosed)
☒ Separate Power of Attorney by Assignee
☐ 37 CFR 3.73(b) Statement by Assignee
☐ Information Disclosure Statement with Form PTO-1449
☐ Copies of IDS Citations
☐ Preliminary Amendment
☒ Return Receipt Postcard
☒ Small Entity Statement(s)
☐ Other:

Claim For Foreign Priority

- ☐ Priority of Application No. filed on is claimed under 35 U.S.C. § 119.
- ☐ The certified copy is enclosed.
- ☐ The certified copy will follow.

- ☐ Amend the specification by inserting before the first line the sentence:

"This application claims the benefit of U.S. Provisional Application No. filed on , the disclosure of which is incorporated herein by reference."

Fee Calculation (37 CFR § 1.16)

	(Col. 1) NO. FILED	(Col. 2) NO. EXTRA	SMALL ENTITY RATE FEE	OR	LARGE ENTITY RATE FEE
BASIC FEE			\$345 \$345	OR	\$690 \$
TOTAL CLAIMS	31 -20 = 11		x 9 = \$ 99	OR	x 18 = \$
INDEP CLAIMS	5 -03 = 2		x 39 = \$ 78	OR	x 78 = \$
[] Multiple Dependent Claim Presented			\$130 = \$	OR	\$260 = \$
* If the difference in Col. 1 is less than zero, enter "0" in Col. 2.			Total \$522	OR	Total \$

- ☒ Check No. 2935 in the amount of \$ 562.00 is enclosed.

- ☒ The Commissioner is authorized to charge any fees beyond the amount enclosed which may be required, or to credit any overpayment, to Deposit Account No. 50-0388 (Order No. BIG1P001).

General Authorization for Petition for Extension of Time (37 CFR §1.136)

- ☒ Applicants hereby make and generally authorize any Petitions for Extensions of Time as may be needed for any subsequent filings. The Commissioner is also authorized to charge any extension fees under 37 CFR §1.17 as may be needed to Deposit Account No. 50-0388 (Order No. BIG1P001).

- ☒ Please send correspondence to the following address:

Customer Number 022434
BEYER WEAVER & THOMAS, LLP
P.O. Box 130
Mountain View, CA 94042-0130
Telephone (510) 843-6200
Fax (510) 843-6203



Date: June 22, 2000

Rupak Nag
Rupak Nag
Registration No. **37,493**
Page 2 of 2

Applicant/Patentee: Joshua A. Jacobs, et al.
Application or Patent No. NEW Atty Docket # BIG1P001
Filed or Issued: HEREWITH

VERIFIED STATEMENT (DECLARATION) CLAIMING SMALL ENTITY STATUS
37 CFR 1.9(f) and 1.27(c)--SMALL BUSINESS CONCERN

I hereby declare that I am
☐ the owner of the small business concern identified below;
☐ an official empowered to act on behalf of the small business concern identified below:

NAME OF CONCERN: Bigstep.com, Inc.
ADDRESS: 530 Hampshire, Suite 400, San Francisco, CA 94110

I hereby declare that the above identified small business concern qualifies as a small business concern as defined in 13 CFR 121.3-18, and reproduced in 37 CFR 1.9(d), for purposes of paying reduced fees under 41(a) and (b) of Title 35, U.S. Code, in that the number of employees of the concern, including those of its affiliates, does not exceed 500 persons. For purposes of this statement, (1) the number of employees of the business concern is the average over the previous fiscal year of the concern of the persons employed on a full-time, part-time or temporary basis during each of the pay periods of the fiscal year, and (2) concerns are affiliates of each other when either, directly or indirectly, one concern controls or has the power to control the other, or a third party or parties controls or has the power to control both.

I hereby declare that rights under contract or law have been conveyed to and remain with the small business concern identified above with regard to the invention entitled: AN INTEGRATED PLATFORM FOR DEVELOPING AND MAINTAINING A DISTRIBUTED MULTIAPPLICATION ONLINE PRESENCE by inventor(s) Joshua A. Jacobs, et al. described in

☒ the specification filed herewith.
☐ Application No. _____ filed _____
☐ patent # _____ issued _____

If the rights held by the above-identified small business concern are not exclusive, each individual, concern or organization having rights to the invention is listed below* and no rights to the invention are held by any person, other than the inventor, who could not qualify as a small business concern under 37 CFR 1.9(d) or by any concern which would not qualify as a small business concern under 37 CFR 1.9(d) or a nonprofit organization under 37 CFR 1.9(e).
*Note: separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities. (37 CFR 1.27)

Name: _____
Address: _____
☐ individual ☐ small business concern ☐ nonprofit organization

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b)).

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further, that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 1001 of Title 18 of the U.S. Code, and that such willful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

NAME OF PERSON SIGNING: Joshua A. Jacobs
TITLE IN ORGANIZATION: Chief Technology Officer

SIGNATURE _____

DATE 6/21/2000

Attorney Docket No. BIG1P001

PATENT APPLICATION

AN INTEGRATED PLATFORM FOR DEVELOPING AND MAINTAINING A DISTRIBUTED MULTIAPPLICATION ONLINE PRESENCE

By Inventors:

Joshua A. Jacobs
175 Bluxome #124
San Francisco, CA 94107
A Citizen of the United States

John F. Shiple
1303 Rhode Island
San Francisco, CA 94107-3246
A Citizen of the United States

Christopher R. Miller
4025 Webster Ave.
Oakland, CA 94609
A Citizen of the United States

Mahesh Tyagarajan
3617 Copperfield Drive, #244
San Jose, CA 95136
A Citizen of India

Kirt David Johnson
1359 De Solo Drive
Pacifica, CA 94044
A Citizen of the United States

Bernard Luke Gallagher
136 27th Street
San Francisco, CA 94110
A Citizen of the United States

Chris Kiernan
233 Gough Street
San Francisco, CA 94102
A Citizen of the United States

Assignee:

Bigstep.com

Entity:

Small

BEYER WEAVER & THOMAS, LLP
P.O. Box 130
Mountain View, CA 94042-0130
Telephone (510) 843-6200

AN INTEGRATED PLATFORM FOR DEVELOPING AND
MAINTAINING A DISTRIBUTED MULTIAPPLICATION
ONLINE PRESENCE

5 **CROSS-REFERENCE TO RELATED APPLICATIONS**

The present application claims priority from U.S. Provisional Patent Application No. 60/142,181 filed July 1, 1999, the entirety of which is incorporated herein by reference for all purposes. The present application also relates to commonly assigned, copending U.S. Patent Application No.

10 09/_____, filed concurrently herewith, for AN EXTENSIBLE DATA MODEL FOR USE IN AN INTEGRATED PLATFORM FOR CREATING A DISTRIBUTED MULTIAPPLICATION ONLINE PRESENCE (Attorney docket no. BIGP002), the disclosure of which is incorporated herein by reference in its entirety for all purposes.

15 **BACKGROUND OF THE INVENTION**

1. FIELD OF THE INVENTION

The present invention relates generally to computer network-based application programming, design, and maintenance. More specifically, it relates to data constructs and methodologies for creating online, multiuser
20 applications based on a single, uniform data model.

2. DISCUSSION OF RELATED ART

As the Internet and the World Wide Web grow and attract more and more viewers and content-providers, there has been a trend towards customization and, more broadly, user-empowerment. This movement in the
25 online arena, whether on the Internet or an enterprise-based network, is marked by enabling users to create, customize and maintain their own

presence on the network. One example of this is the proliferation of customized Web home pages for individuals that contain only information of specific interest to an individual and that can be edited by that individual. A broader example of user-empowerment is giving an online user or entity the ability to create a full-scale, customized Web site. Although users, primarily businesses, have been creating their own Web sites for years, not surprisingly, they have done so by hiring consultants and Web developers to do the work for them. However, as online application development reaches new levels, many of the initial technological and financial barriers are breaking down.

There are now available to online users several tools and wizard programs that allow them to build their own online applications, almost entirely without the help of Internet programmers or Web-site developers. A wizard is a set of steps (presented as separate screens/panels) that automate a task by asking users questions (with between one and ten questions per screen/panel). Many users now create their own home pages or complete Web sites using software from various vendors, such as popular software from Symantec Software Corporation and Microsoft Corporation, to name just two. Another trend that will inevitably grow more in the future is entities, typically commercial entities, building their own Web site to conduct online business or electronic commerce ("e-commerce"). Presently, there are providers that enable these entities to build their own commercial Web site, but there are still a number of drawbacks.

While advances in online application development have allowed users to create their own Web sites, little attention has been given thus far to forming a comprehensive, integrated, or complete solution and to providing a uniform user experience. E-commerce or e-business sites have increased dramatically in complexity, adding functionality such integration with

customer databases, inventory management, and supply-chain management. Presently, no solution provides a platform for the continual advancement and integration of new e-business capabilities into a single framework with common data and user experience. In other words, no single package or toolkit
5 currently available gives a user the ability to create a fully functioning commercially-enabled Web site. Nor do the toolkits provide a uniform user interface or, more broadly, user experience with which the user can become familiar and efficient.

The numerous services needed to build a complete online presence
10 have disparate user experiences, thereby preventing any significant leveraging of skills users gain from a consistent user interface. Presently, users must register with each of the different Web sites where each site provides a different feature or function needed for the user's online presence. For example, a user may use a catalog building application tool from one provider
15 and have to use a reporting or marketing tool from another provider, and then have to address communication between these and other components of the Web site. In another example, a user has to use several different programs such as a Web page development program, a graphics program, a database program, put them together, and then load it up to an ISP. Lack of uniformity
20 in the user experience and the need to go to different providers are major barriers to entities building and maintaining their own online presence.

Although various application development models exist, none are entirely well-suited for enabling an entity to build its own "industrial-strength" network-based application, such as a comprehensive and
25 commercially-enabled Web site. One such model can be referred to as a unified data model or "whiteboards" developed originally at MIT. This model used software agents to gather data and was used for single-application

and multiple users. Many applications developed for online use are inherently multi-application (*e.g.*, creating a catalog, reporting, site-building, etc.).

Another application development model was used with the Newton hand-held digital organizer developed by Apple Computer Corp. The tool developed for

- 5 the Newton enabled the execution of multiple applications but was geared for a single-user space. Naturally, any single-user model is not well suited for online application development. Furthermore, neither model supported a distributed architecture in which various components of the application reside at different places on the network.

- 10 Therefore, it would be desirable to have an integrated platform having a reactive architecture that allows a user to create a distributed, online application that can perform as a complete solution to a goal or problem. The platform should provide for single registration and have a seamless, uniform user experience that fosters leveraging skills learned from previous sessions.
- 15 It would also be desirable for the platform to be distributed, and be multi-user and multi-application. Furthermore, it would be desirable for the architecture to be task-based and to provide for a uniform expression of data, which can be shared and is extensible.

SUMMARY OF THE INVENTION

To achieve the foregoing, software architectures, platforms, and data constructs are disclosed which provide a system for enabling a non-technical
5 or lay user to perform discrete technical tasks necessary to build a complete network-based, multiuser application. The system also allows the user to have a uniform user experience throughout development of the application. For example, the platform can be used to construct and maintain an Internet or online Web site capable of handling e-commerce transactions or can be used
10 to develop a customer relationship management system, two name just two examples.

In one aspect of the invention, a software architecture is described that enables many users to perform a variety of tasks via a wide-area network, such as an enterprise network or the Internet. The architecture has several services,
15 systems, and an extensible database for storing data objects. The database has an underlying structure referred to as a schema that can be extended with previously undefined attributes without having to alter the basic format of the schema. The architecture also includes an integrated platform that enables each of the users to perform the tasks by controlling interaction or
20 communication between the services and systems, and the extensible database.

In one embodiment the software architecture and the platform, in particular, contextualizes each stage of a task according to a particular user. In another embodiment, the architecture and platform allow each user and application to extend the database in a user-specific way. This allows users
25 and applications to use the same database, without having to change the underlying structure of the database. In yet another embodiment, the

architecture provides a user interface or user experience that is uniform across the multiple applications used "behind the scenes" to build the network-based application desired by the user. In yet another embodiment, the software architecture is a reactive architecture which supports various levels of task
5 granularity and is dynamically aware of what information has been entered by a user.

In another aspect of the present invention, an integrated software platform for creating an end-user application having a uniform user experience and a single-registration feature is described. A data model arranges and
10 configures application data, which are either of a fixed attribute type or an extended attribute type. The platform also includes a data logic or dependency logic component for operating on the application data. A back-end code layer manages the uniform user experience and a visual design component implements the user experience primarily by presenting a user
15 interface that is used for entering data into the system. The platform communicates with a data schema or repository for storing the application data configured in the data model.

In another aspect of the present invention, a task-based architecture for building a multiuser, online application through a task-based approach is
20 described. The architecture includes a data schema for storing data related to the online application. A data model stores and shares the data as multiple tasks are completed and is physically stored in the data schema. The task-based architecture also includes multiple tools and a task viewer application for creating a user interface. Several services allow a user to gather and author
25 the data. One of the tools includes a data extension framework for defining and extracting data. Another tool is a context management tool for determining the context in which a particular task is being completed.

In another aspect of the present invention, a system for building a distributed, multiapplication program is described. The system includes multiple tasks, where a task is made up of one or more sequences. The system also includes multiple panel objects within a sequence through which data
5 related to the multiapplicatin program is entered and manipulated. A panel object is aware of data object to acces to retrieve existing application data.

In yet another aspect of the present invention, a method of building a customized Web site is described. One or more Web pages, such as a home page or a dynamic map page is created and maintained. A communication
10 service for communicating with users accessing the Web site is developed. An online transaction system for processing online orders made through the Web site is also developed. A reporting service for generating reports relating to Web site activity is created. The method of building the customized Web site involves a task-based approach to completing an activity and has a
15 uniform user experience.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention will be better understood by reference to the following description taken in conjunction with the accompanying drawings in which:

FIG. 1 is an illustration showing numerous layers of the platform in accordance with one embodiment of the present invention.

FIG. 2 is an illustration of components in a task-based architecture of an integrated platform in accordance with one embodiment of the present invention.

FIG. 3 is an illustration of a data model for an e-commerce enabled Web site in accordance with one embodiment of the present invention.

FIG. 4 is a block diagram of a data object, a model object, and a data schema in accordance with one embodiment of the present invention.

FIG. 5 is a block diagram showing a relationship among tasks, sequences, and panels.

FIG. 6 is a screen illustration showing components of a task viewer application in accordance with one embodiment of the present invention.

FIG. 7 is a block diagram showing various services used for creating and maintaining a commercial online presence in accordance with one embodiment of the present invention.

FIG. 8 is a block diagram showing various tools for use in an application-development system in accordance with one embodiment of the present invention.

FIGS. 9A, 9B, and 9C are flow diagrams illustrating a task-based approach to building a particular application, namely, an e-commerce Web site, in accordance with one embodiment of the present invention.

FIG. 10 is a flow diagram of a process of creating and editing a map page in the application building process in accordance with one embodiment of the present invention.

FIG. 11A is a screenshot showing three map page layout options from
5 which a user can choose.

FIG. 11B is a screenshot showing options available to a user for completing a map page layout task.

FIG. 11C is a screenshot showing how to complete a user direction layout task.

FIG. 10
FIG. 11A
FIG. 11B
FIG. 11C
FIG. 12
FIG. 13
FIG. 14
FIG. 15
FIG. 16
FIG. 17
FIG. 18
FIG. 19
FIG. 20
FIG. 21
FIG. 22
FIG. 23
FIG. 24
FIG. 25
FIG. 26
FIG. 27
FIG. 28
FIG. 29
FIG. 30
FIG. 31
FIG. 32
FIG. 33
FIG. 34
FIG. 35
FIG. 36
FIG. 37
FIG. 38
FIG. 39
FIG. 40
FIG. 41
FIG. 42
FIG. 43
FIG. 44
FIG. 45
FIG. 46
FIG. 47
FIG. 48
FIG. 49
FIG. 50
FIG. 51
FIG. 52
FIG. 53
FIG. 54
FIG. 55
FIG. 56
FIG. 57
FIG. 58
FIG. 59
FIG. 60
FIG. 61
FIG. 62
FIG. 63
FIG. 64
FIG. 65
FIG. 66
FIG. 67
FIG. 68
FIG. 69
FIG. 70
FIG. 71
FIG. 72
FIG. 73
FIG. 74
FIG. 75
FIG. 76
FIG. 77
FIG. 78
FIG. 79
FIG. 80
FIG. 81
FIG. 82
FIG. 83
FIG. 84
FIG. 85
FIG. 86
FIG. 87
FIG. 88
FIG. 89
FIG. 90
FIG. 91
FIG. 92
FIG. 93
FIG. 94
FIG. 95
FIG. 96
FIG. 97
FIG. 98
FIG. 99
FIG. 100

DETAILED DESCRIPTION

Reference will now be made in detail to a preferred embodiment of the invention. An example of the preferred embodiment is illustrated in the accompanying drawings. While the invention will be described in conjunction with a preferred embodiment, it will be understood that it is not intended to limit the invention to one preferred embodiment. To the contrary, it is intended to cover alternatives, modifications, and equivalents as may be included within the spirit and scope of the invention as defined by the appended claims.

10 The present invention describes a platform and data model that allows non-technical users to develop online and network-based applications. The platform can be viewed as an operating system which facilitates self-development of online and stand-alone applications. The data model component, which can also be viewed as a component of the platform, allows
15 for the collection, aggregation, and modeling of user data which leads to efficient utilization and scaling of the data, such that the data can be shared across a high volume of users. The aggregated data, typically from many users, creates a data source which can be shared and leveraged by all the users. The data component or model is also robust in that it is easily extensible,
20 flexible, and uniform.

 The data model provides a foundation for an integrated platform on which a task-based architecture and methodology is implemented. This task-based architecture and methodology for building applications facilitates using a large data pool based on the data model. The data model and task-based
25 architecture also allow for a uniform user experience, as well as other advantages described below. Similar to a conventional operating system, in

order for the integrated platform to function, numerous applications and tools are needed. These applications and tools allow a user to use the data model and the task-based architecture to create a comprehensive, multiuser, online application, such as an e-commerce Web site, a catalog or portfolio-oriented

- 5 Web site (for service-based online businesses), a customer relationship management application which allows a user to manage various types of customer relationships online, marketing and related business-acquisition tools, and different types of customer communication applications. In these types of multiuser, online applications, the user is typically an online merchant
- 10 or small business owner ("SBO") desiring an online presence.

- The integrated platform of the present invention can be described generally as a series of layers, again, much like an operating system. FIG. 1 is an illustration showing numerous layers of an integrated platform in accordance with one embodiment of the present invention. At the bottom of a
- 15 platform 100 is a core data repository or data schema 102 which physically stores data for a one or more data models. A data model, described in greater detail in FIGS. 3 and 4, represents an arrangement or configuration of data in terms of fixed and extensible attributes as well as logic for operating on the data. For purposes of illustrating the described embodiment, data arranged in
- 20 the data model can be, for example, business data and data relating to users, where a user is an SBO or online merchant. The data model has a unique configuration that facilitates the collection and aggregation of data and the addition of previously undefined data attributes, referred to as extensibility. Data schema 102 physically stores the data used by the applications, tools, and
- 25 functions in conjunction with the task-based architecture. Data schema 102 can store data in the form of a relational database, flat files, a multidimensional database, or any other appropriate data storage schema.

Immediately above data schema 102 is a back-end code application layer 104. In the described embodiment, application layer 104 contains numerous application modules and programs that run behind the scenes of the user experience; that is, what the user sees and does when using the integrated platform. Above back-end code application layer 104 is an interface layer 106 which is the first layer of the system front-end. In the described embodiment, the front-end, consisting of interface layer 106, an information architecture layer 108, and a visual design layer 110, is loosely tied to back-end code application layer 104. Interface layer 106 provides a container mechanism, usually implemented as an HTML form element, which wraps data for transfer between application layer 104 and interface layer 106. Interface layer 106 connects layer 104 with an information architecture layer 108 and a visual design layer 110.

Information architecture layer 108 maps a user's "mental model," *i.e.*, the user's knowledge domain, to data repository 102. Layer 108 also maps the user's mental model of a task-based interface to interface layer 106 and application layer 104. Information architecture layer 108 models the user experience and embodies a task-based architecture by implementing and reinforcing various models and flows for each task within the system, described below.

Interface layer 106 and application layer 104 translate the instructions provided by information architecture layer 108 and present the desired screens/panels to the user. Through this process, the technicalities and expertise normally needed to complete otherwise highly technical tasks are abstracted away. A visual design layer 110 rests on top of information architecture layer 108 and implements a user experience by presenting the

actual interface that the user uses to enter data into the system. Visual design layer 110 implements the user interface for the task-based architecture.

Layers of integrated platform 100 that are part of the task-based architecture and, thus, particularly relevant are information architecture layer 108, back-end code application layer 104, and data schema 102, or, more specifically, the data model embodied in data schema 102. A more detailed view of the last two layers is shown in FIG. 2.

FIG. 2 is an illustration of components in a task-based architecture of an integrated platform in accordance with one embodiment of the present invention. At the bottom is data schema 102 which physically stores the data. Operating on data schema 102 is a data model 202 that configures and handles data stored in data schema 102. As mentioned above, a data model facilitates collecting, storing, and sharing information from a user. The data are modeled in such a way that they allow for efficient reuse of data already entered, in a way that is intuitive to and expected by the user. Data arranged in data model 202 are dynamically configured as the user completes various tasks. Data are expressed uniformly and can be shared by disparate and newly added applications and modules in the system. In the described embodiment, data model 202 has an Extensible Markup Language (XML) layer 203 that further facilitates the flexibility of data model 202 and its ability to collect and output data in a distributed manner. Thus, data model 202 allows for the abstraction, aggregation, and modeling of data in a common integrated platform and enables the task-based architecture of the present invention. Data model 202 is described in greater detail in FIG. 3.

In direct communication with data model 202 is a toolkit layer 204 containing numerous toolkits 206. Toolkits 206 are modules, systems, and frameworks that perform core functions of the application-building system of

the present invention. The toolkits necessary for implementing the task-based architecture of the integrated platform are described in FIG. 8.

Operating in conjunction with toolkit layer 204 is a task viewer application layer 208. The application-building system of the present invention can be implemented by a service provider who manages the integrated platform, the data model, data repository, and other components of the system. The application-building system is a collection of database-driven applications. One such application is a task viewer application. A task viewer implements an interface for users to create relevant data, such as business data for an SBO. The task viewer provides a core set of services enabling a uniform user experience. These services include the delivery of sequential forms and screens for gathering and formatting information, information validation, and information storing services. The primary responsibility of the task viewer is the delivery of an atomic data gathering procedures. In order to provide flexibility in authoring data, these procedures have been partitioned into three discrete types of objects: tasks, sequences, and panels, each with their own function, described in FIG. 5. The task viewer application operates as a programmable wizard framework. In order to take advantage of the extensible data model of the described embodiment, an authoring system allows data objects in the data model to be created, extended, and stored by the system.

A services layer 210 can be considered part of interface layer 106 and the front-end of the application-building system. In the described embodiment, there are various services 212 that allow a user to create a complete, multiuser application without having to use any external programs or services. As mentioned above, one such multiuser application is creating a comprehensive e-commerce Web site for a user who wants an online presence.

In such an application, services 212 refer to defined functions typically needed in order to fully complete the application. In the customized e-commerce Web site application of the described embodiment, the following broadly defined services are typically needed: 1) site building; 2) communication; 3) catalog or portfolio; 4) commerce; 5) marketing; and 6) reporting. In another preferred embodiment, a Web site-building application can have more or fewer services as the needs of Web sites can vary over time. In yet another preferred embodiment, services of a different nature or type can be used for a different application, such as a customer marketing campaign.

FIG. 3 is an illustration of a data model for an e-commerce enabled Web site in accordance with one embodiment of the present invention. At the center of a data model 300 is a shop system 302. Shop system 302, as with other systems described below, is a collection of model objects. Shop 302 contains a unique sequence identifier ("SID") used for identifying a user.

The SID is used by other systems to key into shop 302. When a user logs in, the user's SID is retrieved and used to pull in the rest of the data from the other systems for that user. Each of the systems can get to shop using the user's SID. Shop 302 is the key to getting data from the other systems.

A task management system 304 is a wizard framework for handling all user interaction through tasks, sequences, and panels. These specific components are described below. Task management system 304 is responsible for presenting a standard, seamless, user interface to a user for all data creation and editing functions. It allows for tracking and resuming tasks when a user stops working on a task before the task is completed. It is also responsible for the identification, verification, and publication of all data it receives from a user through panels.

A partner system 306 is used primarily by the application-building service provider and not the user. It allows the service provider to partner with third-parties through which users choose to build their applications. Partner system 306 is an aggregator of objects that relate to aspects of the application-building service that can be modified according to a particular partner. It is an extensible mechanism for providing business term-specific overrides to the integrated platform at runtime. For example, it contains models objects and data objects that control aspects of the user experience, such as color, fonts, and partner logos, that can be customized for a particular partner of the service provider. Thus, these implementation-specific variables can change depending on the partner and are modeled and loaded into the environment by partner system 306. This is done without assistance from or effecting the other components or systems in the platform.

A consumer system 308 maintains a database of end-users somehow associated with the user, such as a list of visitors and customers to an SBO Web site. This list or database is fully integrated and addressable from other systems in data model 300 using the user's SID. A site management system 310 encapsulates data relating to a user's Web site. A user can have multiple sites, such as one that is being created and is previewed only by the user and another that is published and viewable by the public. A commerce/catalog system 312 is a catalog database attached to a user through shop 302, similar to the database in consumer system 308. A typical catalog database contains descriptions and prices of products and services available from the user. Additionally, this system is responsible for recording all commerce transaction associated with the shop. This includes an order management and processing function which is tied to consumer system 308. A user management system 314 performs general user authentication and checks

roles and permission levels within the system. For example, a user, such as an SBO, may have several employees who can access the application-building service, some of whom can perform only certain tasks, such as creating a map page, but not others, such as establishing a merchant account. User

5 management system 314 controls access of users in the system.

Also shown in data model 300 are new system or application constructs 316A and 316B. These constructs show that new systems and applications can be added to data model 300 dynamically without effecting existing systems. As long as new systems 316A and 316B can be accessed using a user's SID, the platform can be extended to include them. This can be done by creating additional model objects and data objects, described in FIG. 4, and without altering the structure of the underlying data schema. In addition, new systems 316A and 316B can use data from the other systems described above.

15 FIG. 4 is a block diagram of a data object, a model object, and a data schema in accordance with one embodiment of the present invention. A data object 400 is used for efficient persistent storage and is capable of reading from and writing to data schema 102 but is not capable of performing any logical operations on the data. In the described embodiment, data object 400 has two components: fixed attributes 402 and extensible attributes 404. Fixed attributes 402 represent items known to the user at the time the data object is created and may include data items such as a URL or Web address, a user name, address, and other basic data. Some data items can be divided into subcomponents. For example, phone number can be stored as several

20 components: country code, suffix, prefix, area code, and so on.

25 Extensible attributes 404 include data that are user-specific or are added to the user system at a later time or cannot be defined in fixed attributes

402. Extensible attributes 404 can include user-specific preferences, such as performing a task in beginner or advanced mode, or hiding or showing a summary area for a task. Another example of extensible attributes is "Frequently Asked Questions" or FAQs data for an SBO which are typically very business specific and, therefore, ideal for being stored as extensible attributes in a data object. Arrow 406 represents that data of either attribute can be shared, or can be converted from one type to another when allowed by the system.

Using an example of an e-commerce enabled Web site, an SBO or online merchant can be represented as a collection of data objects 400. These data can be further reduced into categories that represent, at a high level, components of the business. To illustrate, business data can generally be categorized into four areas: business information (*e.g.*, hours of operation, location, name, tax information, etc.), a catalog of good and/or services which the business offers, a collection of customers and their related attributes (*e.g.*, name, contact information, profile information, billing, etc.) and financial data such as income and operating costs. The information is collected via applications in these categories and is utilized to deploy a core set of user applications that rely on this data for functionality. Information is entered, exchanged, and reused among components in a multiuser application. The e-commerce Web site is described as a collection of data. The model for this data can be populated by many applications. Examples of multiuser applications include the creation of Web pages, management of a complex Web site, construction of online catalogs or portfolios, as well as online order processing functionality, and e-commerce transactions.

In order to manage information related to these categories, the data are categorized into a collection of objects that encapsulate and manipulate the

data. These objects are referred to as model objects. A model object 408 is a containment structure for data objects through which all interaction between applications and the data schema occurs. It contains business intelligence in the form of dependency logic 410 for one or more related data objects 412.

- 5 Dependency logic 410 understands the relationship between data objects 412 and knows what needs to be done when certain data are modified. In a simple example, when a user changes a Web address, at the data object level, a URL field in fixed attributes component 402 is written to with the new address. However, data object 400 does not know that an external DNS server needs to be notified. Model object 408, and specifically, dependency logic 410, understands that the DNS server needs to be notified and that there needs to be a storage change in the data schema, and causes these events to occur.

- 10 Data schema 102 is the physical storage of data model 202 which, in turn, is a collection of data objects. An API 414 between model object 408 and data schema 102 allows for modification of the data. API 414 is determined by the type of physical data storage mechanism used, such as a relational database or a multidimensional database.

- Besides being a container of data objects, model object 408 provides a user with a standardized interface for manipulating attributes in data objects
- 20 412. Users do not need an external understanding of the data objects' interdependencies or how the data are configured and stored. In sum, the model object can be seen as a type of logical interface between the application-building system and the user, and the data objects, as contained in model objects, as a type of database interface. In the described embodiment,
- 25 model objects and data objects are implemented by encapsulating them using Java™ bean technology as is known in the field of Internet and network-based application programming.

In order for the integrated platform to process the disparity among different types of business or non-business data, the platform requires a mechanism for extending the attributes of a given model object. As mentioned above, in the described embodiment, XML is used to implement data and logic for the model objects and data objects. As is known in the field of Internet application programming, XML allows for arbitrary attributes to be structured in a hierarchical format. This hierarchical format enforces structure and dependency while allowing for changes that do not affect external clients or users of the contained data. Thus, data object 400 exposes its data to model object 408 via an XML layer (shown as XML layer 203 of FIG. 2) thereby allowing for attributes to be added as desired, *i.e.*, making them extensible. Model objects can be extended as new relationships form which require enforcement of data dependencies.

The interaction between a model object and data object can be illustrated by taking the example of a model object that acts on a relationship between an SBO and an online customer. The model object in this case enforces a dependency binding a data object to a business entity. Extensible attributes of such a data object may begin with data relating to when the relationship was established, how the customer was referred to the SBO online site, and whether a transaction was completed. Using XML to define these attributes facilitates extending the system *i.e.*, the SBO's Web site, to be aware of the number of return visits by the customer without requiring the SBO to have an understanding of or having to modify the underlying storage structures or dependencies. Over time, the XML definition for such a "customer relationship" object may expand to cover all aspects of the history of transactions between the business and the customer. By virtue of XML, the original specification describing when the customer first visited the Web site

remains unchanged while whole new subcategories of information have been added to the object. The overall model has added no new rules to support the new data, nor has the underlying data layer and storage system changed to support the potentially vast data contained by the system.

5 FIG. 5 is a block diagram showing a relationship among tasks, sequences, and panels. A task 500 is a goal-based activity or operation that a user performs in the application-building process. One example is creating a home page in the case of building an e-commerce Web site. Another example is creating an ad banner, generating a list of potential consumers, and enabling
10 the ad banner to be presented to the potential consumers in a consumer marketing campaign application. Tasks are managed by a task viewer in task viewer application layer 208 of FIG. 2 and further described in FIG. 6. Task 500 is made up of one or more sequences 502, where a sequence is geared towards completing a more granular operation within the task, such as
15 selecting an overall layout of a home page. The number of sequences in a task can vary over time, as can the number of tasks in the application-building process. A sequence, in turn, is made up of a series of panel objects 504. It is at the panel level where the user typically creates or edits data. Examples of this are shown in FIGS. 9 to 12.

20 Each panel object 506 represents an atomic step in task 500. Panels 504 do this by creating form elements that enable windows for data to be entered, displayed, or edited. A form element is an HTML construct that defines the fields that will be sent to a server when a user clicks a SUBMIT button. A panel object 506 is responsible for knowing where to go in the data
25 objects to get its default data. This default data are used for describing the requirements for valid edits of the data and for describing where in the system to store valid results. Panel object 506 can describe valid inputs for each of

their form elements or windows. Validation methods are used to determine if a panel object's contents meet the criteria for storage in one or more data objects. The task viewer manages validation by refusing to advance past panels that do not meet the panel object's validation criteria.

5 The user experience of the application-building process of the present invention is based on a task-by-task approach. In this model, users complete tasks (self-guided, goal-based operations) to build and manage various parts of their application, such as an e-commerce Web site, including the creation of Web pages, management of a potentially complex Web site, construction of
10 online catalogs, online order processing functionality, and e-commerce merchant functions. The architecture is designed to have discrete layers of functionality, building up from discrete actions or panels to combinations of panels to form sequences which, in turn, form tasks. Discrete actions, such as filling out a form field or clicking a button, are combined together to make a
15 task such as building a Web page. Discrete tasks combine together to form an activity, such as creating and managing an online catalog. An example of a task-based approach to building an application is shown in FIGS. 9A, 9B, and 9C.

FIG. 6 is a screen illustration showing components of a task viewer
20 application in accordance with one embodiment of the present invention. A task viewer application 600 is implemented as a collection of JHTML (Java™ HTML) files. Task viewer 600, as displayed in a user interface, has numerous sections: a controls area 602 for the application; a current task information area 604; navigation elements 606; a configurable HTML form region 608
25 which allows submitting/viewing information; and a content region 610 acting as a placeholder for contents for a current panel object.

FIG. 7 is a block diagram showing various services used for creating and maintaining a commercial online presence 700 in accordance with one embodiment of the present invention. A site building service 702 allows a user to build and maintain Web pages in a step-by-step manner in which discrete tasks are completed. This service gives the user control over the site's appearance and content, and allows the user to make changes to the Web site at anytime. This is enabled in large part by data model 202. A communication service 704 facilitates communicating with online users viewing the online merchant's site through such services as personalized email newsletters and online surveys. Communication service 704 can also facilitate building a customer database storing customer buying patterns and personal preferences, all of which can improve communication between the online merchant and visitors to the site.

A catalog service 706 allows a user to display products or services through an online catalog or portfolio. The user can dynamically change the format, style, and information. Given that an online e-commerce site is being created, a merchant account and an online transaction system needs to be in place. This can be done using a commerce service 708. This service allows users to manage online orders and establish an efficient checkout process, if needed. A marketing service 710 lets the user promote the online business using various marketing tools. For example, marketing service 710 facilitates submitting the site to search engines. As with the communication service, the marketing service takes advantage of email newsletters and surveys, as well as banner advertising. A reporting service 712 allows the SBO to create sales and customer profile reports. Such reports can also include statistics about Web site activity and trends in visitor traffic.

FIG. 8 is a block diagram showing various tools for use in an application-development system in accordance with one embodiment of the present invention. It describes in greater detail toolkit layer 204 and toolkits 206 of FIG. 2 describing the task-based architecture of the present invention.

- 5 In the described embodiment, a contextual template system (CTS) tool or context manager 802 used to filter all derived objects based on a current user state. CTS implements a registry of extensible context checks and uses them to maintain a polled snapshot of the current operating environment. This context signature is used to retrieve certain records (e.g., Task, Help, and
- 10 Renderer records) from the database, and exclude or prioritize them based on their self-described contextual requirements. The CTS or context manager is a runtime object that is responsible for maintaining the context signature for a current user, and is used by other objects to retrieve a list of relevant objects.

- In the described embodiment, there are three components in CTS 802.
- 15 Collectively, they are responsible for the management of contextual information in the system. A context check component 804 is used by context manager 802 to assess and maintain the context signature. Context check module 804 is a module of code which produces a Boolean result. Context checks are written to allow for filtering of a particular rule. An example of this
- 20 might be a check that is written to determine if the current user is "owned" by a specific partner of the service provider. Certain Renderers (e.g. page layouts) might only be allowed for this specific partner, in which case the task would specify a "Requires" context check for that specific aggregator. In this context a partner is a third-party that has co-branded the service provider's
- 25 interface and offers it to their users. Such partners may wish to add or subtract various functionality within the application-building process. In another example, a user can be in a basic or advanced mode within a task or the user

may be in a create sequence or an edit sequence. For many tasks, users are walked through a create sequence to set a Web page or create their customer database. After this, the user can access data that they have created within edit mode or sequence thereby avoiding having to walk through all the steps again.

5 Another component is a context information record 806. All contextual objects include a context information record. In the described embodiment, context information records have three sections: Requires, ExcludeIf, and MandatoryIf. Each of these categories is expressed in record 806 as a list of context check IDs and the desired value for the condition to be true. In a natural language example, a task might express itself as follows: I Require that the user has already built a site, but I should be ExcludedIf the customer has already added a map to their site.

Context Check module 804 is programmed using any Java™ development environment. The resulting class file is registered with a name and ID using a content registration tool 808. This tool is able to list context check modules by name when Tasks and Renderers are being registered.

Context Checks are stored in the database and registered when CTS 802 is constructed. CTS 802 manages a runtime instance of each context check. At pre-determined times the CTS polls each of the checks to produce a current signature. When asked for a list of contextual objects, the CTS uses the current signature to filter and sort the results of the database query, resulting in a qualified list of Tasks or Renderers.

An Extended Attribute Framework (EAF) 810 is a tool by which data object attributes are defined, manipulated, and extracted from data schema 102. This attribute framework is extended by all classes in the system which support attributes. Framework 810 provides interfaces and methods which need to be implemented by the extending classes. The attributes themselves,

for each instance of a class extending EAF 810, are persistent and present in the database.

In the described embodiment, EAF 810 consists of an "attributecollection" class 812 which consists of attribute manipulation methods. The manipulation of attributes includes methods to add, remove, replace and get attributes. EAF 810 implements an indexed hierarchical collection of attributes. Each attribute has associated with it a name, type and index within the collection.

Another tool is a task management system ("TMS") 814 system which is used to deliver a task-based interface. Task management system 814 encapsulates the design and storage of task sequences 502, their related panel objects 504, their in-memory representation, and their interaction with task viewer application 600. Task manager 814 consists of four components that collectively manage the design, delivery, and storage of task information.

A task list component 816 is a persistent object with two primary purposes. First, it maintains a list of tasks that customers have specified an interest in starting at a deferred time. Second, task list component 816 is used to determine a context-sensitive suggestion of tasks that the current customer might want to pursue.

Task viewer component 818 defines a user interface shown generally in FIG. 6. It is a JHTML based application that interacts with the run-time object representing a currently active task. Task viewer 818 includes a collection of navigation controls (*e.g.*, next, previous, and special), a title area, a content region which contains the current panel in the sequence (containing form/data entry fields), and a collection of task management widgets such as Save buttons. Task viewer application 818 queries the current task to determine the next panel to display, the appropriateness of specific navigation

elements, the title to display, and other information relevant to the accomplishment of the task. As such, the in-memory representation of a task object is a model which is manipulated by task viewer 818 acting as controller.

- 5 A task 822 is a read-only persistent object that encapsulates visible aspects of a task, contextual information for determining the task's relevance under dynamic conditions, and the sequence logic required for its delivery. In the database, tasks are delivered as an attribute list with high-level accessors available for common functions. As described above, a panel 820 is an atomic
10 element of a sequence that is used to perform a specific step in a task. Panels are registered in the system to provide for reference counting of the sequences that contain them, and the management of their assets.

- Task viewer application 818 is launched in response to a request to perform a task. A task 822 is either user-added or dynamically added to task
15 list 816. When a task is started a TSPTask object is added to a current session object. The TSPTask acts as the intermediary for identifying how task viewer 818 should respond to user interaction events.

- FIGS. 9A, 9B, and 9C are flow diagrams illustrating a task-based approach to building a particular application, namely, an e-commerce Web
20 site, in accordance with one embodiment of the present invention. The process of building such an application is shown in three sample phases. Phase 1, illustrated in FIG. 9A, shows generic tasks for developing an e-commerce Web site. Phase 2, illustrated in FIG. 9B, shows sample tasks needed for building an online catalog to be used in the Web site and Phase 3,
25 illustrated in FIG. 9C, shows sample tasks for activating e-commerce functions.

At step 902 a user, such as an SBO, takes on the first task of registering a domain name. In the described embodiment, a user can either register their existing Web address (e.g., www.my-store.com) or register a domain name with the application-building service provider, in which case the address would resemble www.my-store.service-provider.com. In the later case, the service provider verifies that the name for "my-store" is available and verifies basic business information such as address and other user contact information, including the user's email address. At this stage the user also creates a password for his account. It should be noted that the entire application-building process requires that the user only register once with the service provider. The user does not have to re-register or register multiple times with different application tools when beginning work on a different aspect or feature of the application.

At step 904 the user performs the task of determining the general design and look of the Web site by selecting, for example, a color scheme, design layout, and character fonts of the site. In the described embodiment, the display aspect of the site is separated from the content. In this task, the display aspect is being determined. By doing so, the user is settling on a consistent look and appearance for the site. In the described embodiment, this is done by offering the user a number of options or templates for layout designs and color schemes. Each template is a set of HTML objects and graphic templates, and is coded to contain, for example, predetermined fonts, foreground and background colors, and other graphical features. By choosing a template the system can dynamically populate the display.

Once the user has completed this task, at step 906 she creates and edits a top-level page or home page. Typically, a home page introduces a consumer to the user's business and may describe generally the goods or services being

offered. The home page is typically updated frequently by the user to reflect, for example, changes in the user's business (*i.e.*, new products or services) or contact information. At step 908 the user completes the task of creating and editing a map page. The various panels, or steps, for completing this task are
5 described in greater detail in FIG. 10. At step 910 the user creates and edits a customer contact page which essentially displays more detailed information about contacting the user, which can be useful if the user is a large, nation-wide business with many locations and departments. These last two tasks are only two illustrative examples of pages that the user can create for the Web
10 site. In the described example, templates are provided to the user to easily create and edit these pages. Some of these templates are shown in FIGS. 11A, 11B, and 11C for creating and editing a map page. Numerous other templated pages can be offered to a user: an "About Us" page, a "Business Relationships" page, a "Job Listings" page, "Frequently Asked Questions,"
15 "Employee Bios," and so on. Also in the described embodiment, the user can create any number of customized Web pages for displaying information that suits the user's particular needs.

Steps 902 to 910 illustrate five examples of tasks for creating a basic commercially viable Web site which, in turn, is one example of a multiuser
20 application. In other embodiments, numerous other types of tasks may be more appropriate for a particular application or for the site-building application. As mentioned above, these tasks can be grouped together as phase 1 of the application-building process. It should also be noted that these tasks and the ones following are presented to the user in a consistent manner.
25 The more tasks the user completes, the more comfortable and efficient the user becomes with the process, which does not change significantly as the user

advances in the process. This enhances the overall user experience in using the application-building process of the present invention.

FIG. 9B illustrates an example of a second phase of the process. Here the tasks are oriented towards a more specific aspect of the Web site-building application. The example described is building an online catalog for the user's business. Most SBOs, for example, will want to display to consumers a catalog or list of goods and services offered along with photos, descriptions, prices, and so on. At step 912 the user creates category pages. This task is useful for categorizing and distinguishing the user's various goods and services. For example, if the user is an SBO in the retail wine business, there could be a category page for Reds, Whites, Dessert, and so on, and additional category pages within each category.

The next described task is creating and editing items to be listed in the catalog, as shown in step 914. At step 916 the user can upload photos or pictures of the items shown in the online catalog. At step 918 the user creates and edits a front page for the catalog that can, for example, generally describe to consumers the various categories of goods or services being offered. Another possible task involved in building an online catalog is rearranging the order of items on the category pages at step 920. This completes one example of a phase 2 series of tasks for completing another aspect of a Web-site application building process.

FIG. 9C describes yet another series of tasks for developing a commercially viable Web site. It describes tasks for activating certain e-commerce functions normally needed by businesses for accepting and processing actual orders online in which consumers can make payments, for example, with a credit card and have payments go directly to a merchant account. At step 922 the user applies for an online merchant account for

accepting payments from consumers. The service provider has an API to a third-party partner specializing in merchant banking and related online shopping cart services. As part of this task, the user completes an online form as required by the third-party, and the service provider then submits this to the third-party. In another preferred embodiment, the user can open an online merchant account prior to beginning the application-building process and enter the existing information, such as bank name, account number, in place of performing this task (entering such existing information would itself be considered a task).

At step 924 the user prepares a series of shopping cart checkout messages. These are messages displayed to a consumer during the whole process of making a purchase online. In a typical example, such messages can begin when the consumer first places a first item in his shopping cart and can end with the content of an email to the consumer when the goods are shipped out. The user can choose from standard messages, such as "Thank you for your order . . ." to more customized messages. For example, the user can select whether the confirmation email to the consumer should contain a list of the items purchased or a simpler message. At step 926 the user fills out a shipping rates table to inform consumers of shipping costs and factors effecting such costs such as weight, number of items, etc. Similarly, at step 928 the user sets up a sales tax table.

At step 930 the user prepares a merchant policy stating practices and policies the merchant abides by in processing online transactions. In the described embodiment, the user, if an SBO or other type of merchant, may be required to abide by a set of standards for issues like refunds, shipping, and so on, considered to be good e-business practices. At step 932 it is determined whether the merchant account applied for in step 922 has been approved. If it

has not been approved, the user can contact the service provider at step 934. If it has, the described example of activating e-commerce functions is complete and the SBO or online merchant can begin processing online transactions. As with the previous phases, this is just one example of a series of tasks that might be needed for building an e-commerce Web site.

FIG. 10 is a flow diagram of a process of creating and editing a map page in the application building process in accordance with one embodiment of the present invention. It describes the task shown in step 908 of FIG. 9A. In the site-building application, the user can create a customized, interactive map page for the site. The user's business address is plotted on the map and consumers can zoom in, zoom out, or recenter the map as desired. At step 1002 the user completes a panel for creating an introduction panel for the map page. In the described embodiment, the introduction panel informs consumers of a physical location of the user's business, including directions and other information. At step 1004 it is determined whether the user has previously entered his address. If not, the user does so at step 1006. Once the user's address is entered, the user chooses a layout panel for the map page at step 1006. Here the user can select from several different layouts. FIG. 11A is a screenshot showing three map page layout options from which a user can choose. As shown in panel 1102, the user selects one of three layouts: layout 1104, layout 1106, or layout 1108, as part of completing this particular task.

At step 1010 the user specifies a map panel which determines how the map will be displayed the first time a customer views it. Customers can then interact with the map by zooming in or out or recentering the map. FIG. 11B is a screenshot showing options available to a user for completing this task. In a panel 1110 is an initial map layout 1112. The user can re-orient layout 1112 by selecting one of the direction buttons, such as buttons 1114 for SW and S.

The user can select the level of desired detail by using a slider 1116 for zooming in or out.

At step 1012 the user completes a panel for specifying a directions display area. Here a user can provide directions to consumers. FIG. 11C is a screenshot showing how this panel is completed. In a panel 1118 is map layout 1112 and numerous text entry boxes that are self-explanatory. Box 1120 allows a user to enter a headline for the directions, box 1122 allows the user to enter specific directions for consumers, and box 1124 lets the user add any additional information that might be useful to a consumer, such as parking information. At step 1014 the user can edit the map page by bringing up a screen similar to FIG. 11C and editing the text in boxes 1120, 1122, and 1124.

Although the foregoing invention has been described in some detail for purposes of clarity of understanding, it will be apparent that certain changes and modifications may be practiced within the scope of the appended claims. Furthermore, it should be noted that there are alternative ways of implementing both the process and apparatus of the present invention. Accordingly, the present embodiments are to be considered as illustrative and not restrictive, and the invention is not to be limited to the details given herein, but may be modified within the scope and equivalents of the appended claims.

CLAIMS

What is claimed is:

- 5 1. A software architecture for enabling multiple users to perform a plurality of tasks via a wide-area network, the software architecture comprising:
 - a plurality of applications;
 - a data schema for storing a plurality of data objects, the data schema
 - 10 having an extensible, underlying data model; and
 - an integrated platform for enabling each of the multiple users to perform at least one of the tasks by controlling interaction between two or more of the applications and the extensible data model.
- 15 2. A software architecture as recited in claim 1 wherein the integrated platform dynamically contextualizes each stage of a task with reference to a corresponding user.
3. A software architecture as recited in claim 2 wherein the integrated
- 20 platform creates a contextual task list for the corresponding user.
4. A software architecture as recited in claim 1 wherein a data object is associated with one or more contextual describers that further describe a task in which the data object will be used.
- 25 5. A software architecture as recited in claim 1 wherein the platform allows a user and an application to extend the database in a user-specific way, thereby enabling the multiple users and the plurality of applications to use the same database.
- 30 6. A software architecture as recited in claim 5 wherein the database can be extended dynamically without changing the underlying structure of the database.
- 35 7. A software architecture as recited in claim 1 wherein the data model has extensible attributes that can be used to add a previously undefined data

attribute and wherein the extensible, underlying data model provides a standard way of representing the previously undefined data attribute.

8. A software architecture as recited in claim 1 further comprising a user
5 interface that is uniform across the plurality of applications.

9. A software architecture as recited in claim 1 wherein additional services can be added using the plurality of applications.

10 10. A software architecture as recited in claim 1 wherein the integrated platform is used to create and maintain an online business presence.

11. A software architecture as recited in claim 1 wherein the integrated platform is used to create and maintain a customer relationship management
15 application.

12. A software architecture as recited in claim 1 wherein the architecture is a reactive architecture which supports a plurality of levels of task granularity and is dynamically aware of what information has been entered by a user.

20 13. An integrated software platform for creating a user application having a user experience comprising:

a data model for arranging and configuring application data, wherein the application data are one of a fixed attribute and an extended attribute;
25 a data logic component for operating on the data;
a back-end code layer for managing the user experience; and
a visual design component for implementing the user experience by presenting a user interface for entering data into a computer system.

30 14. An integrated software platform as recited in claim 13 further including an information architecture layer for modeling the user experience.

15. An integrated software platform as recited in claim 13 wherein the user application is a multiuser, online application.

35

16. An integrated software platform as recited in claim 13 wherein the user application is a customer relationship management application.
17. An integrated software platform as recited in claim 13 further
5 including a data schema for storing data configured in the data model.
18. An integrated software platform as recited in claim 17 wherein the data schema is implemented as a relational database.
- 10 19. An integrated software platform as recited in claim 14 further including an interface layer for containing a plurality of HTML form elements.
20. An integrated software platform as recited in claim 19 further
15 including an information layer, wherein the information layer and the back-end code layer translate instructions from the information architecture layer.
21. A task-based architecture for building a multiuser, online application by completing a plurality of tasks comprising:
20 a data schema for storing data related to the online application;
a data model for storing and sharing the data as the plurality of tasks is completed;
a plurality of tools;
a task viewer application for creating a user interface; and
25 a plurality of services for gathering and authoring the data.
22. A task-based architecture as recited in claim 21 wherein the plurality of tools includes a data extension framework for defining and extracting data.
- 30 23. A task-based architecture as recited in claim 21 wherein the plurality of tools includes a context management tool for determining a context in the user-oriented application development system.
24. A system for building a distributed, multi-application program
35 comprising:
a plurality of tasks;

one or more sequences within a task;
a plurality of data objects; and
one or more panel objects within a sequence through which data
related to the multi-application program is entered and manipulated, wherein a
5 panel object is aware of which data object from the plurality of data objects to
access to retrieve existing data related to the multi-application program.

25. A system as recited in claim 24 further comprising a plurality of model
objects, a model object containing one or more data objects and a logic
10 component for operating on the one or more data objects.

26. A method of building a customized Web site comprising:
creating and maintaining one or more Web pages;
developing a communication service with users accessing the Web
15 site;
developing an online transaction system for processing online orders
made through the Web site; and
creating a reporting service for generating reports relating to Web site
activity, wherein the method of building the customized Web site includes a
20 task-based approach to completing an activity and has a uniform user
experience.

27. A method as recited in claim 26 further including creating and
maintaining a catalog for describing one of one or more products and one or
25 more services.

28. A method as recited in claim 26 wherein the task-based approach
further includes completing one or more sequences, a sequence including one
or more panels.

30 29. A method as recited in claim 26 wherein creating and maintaining one
or more Web pages further includes controlling the appearance and content of
a Web page.

30. A method as recited in claim 26 wherein developing a communication service with users further includes developing a user database according to user behavior patterns and preferences.

- 5 31. A method as recited in claim 26 wherein developing an online transaction system for processing online orders further includes establishing an online account and checkout process.

10

11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2

ABSTRACT OF THE DISCLOSURE

Software architectures, platforms, and data constructs are disclosed which provide a system for enabling a non-technical or lay user to perform discrete technical tasks necessary to build a complete network-based, multiuser application. The system also allows the user to have a uniform user experience throughout development of the application. For example, the platform can be used to construct and maintain an Internet or online Web site capable of handling e-commerce transactions or can be used to develop a customer relationship management system. A software architecture is described that enables many users to perform a variety of tasks via a wide-area network, such as an enterprise network or the Internet. The architecture has several services, systems, and an extensible database for storing data objects. The database has an underlying structure referred to as a schema that can be extended with previously undefined attributes without having to alter the basic format of the schema. The architecture also includes an integrated platform that enables each of the users to perform the tasks by controlling interaction or communication between the services and systems, and the extensible database.

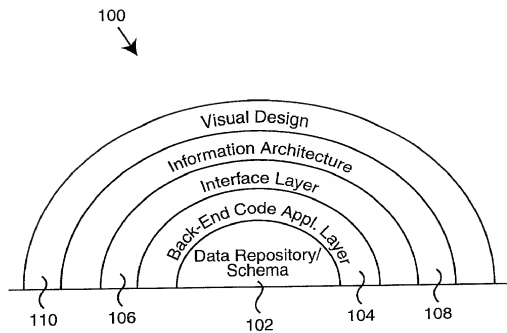


FIG. 1

DATA MODEL 300

FIG. 3

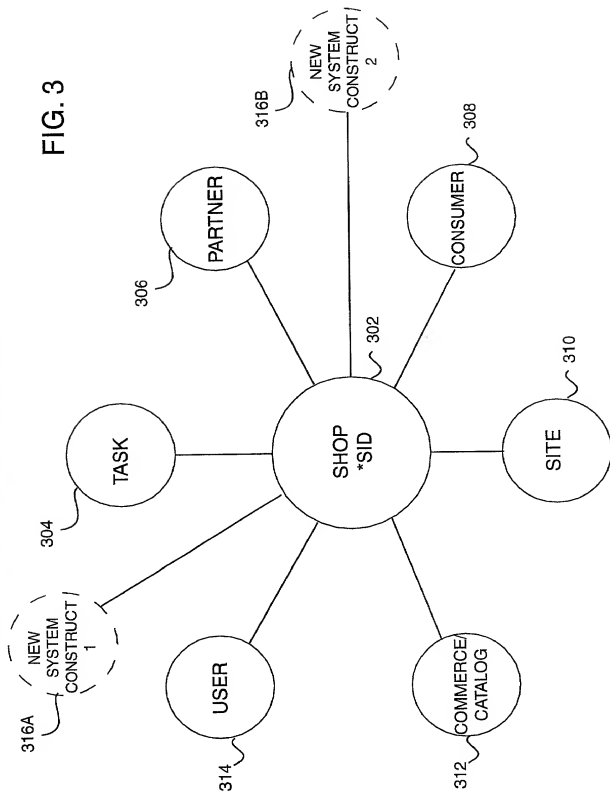
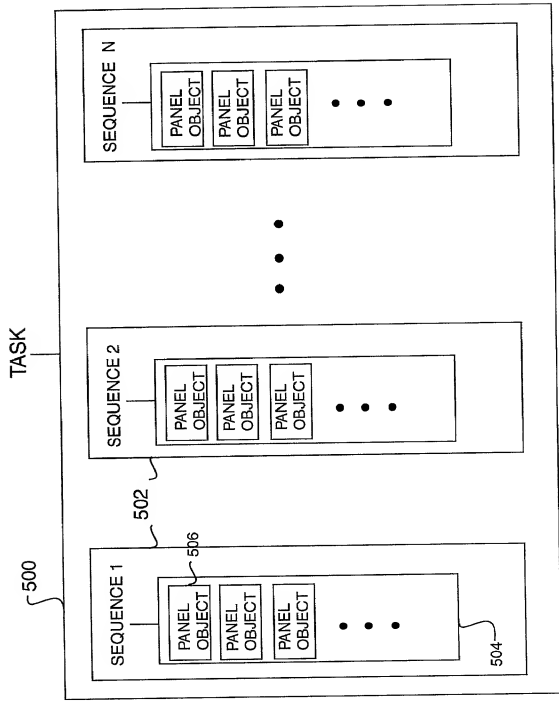


FIG. 5



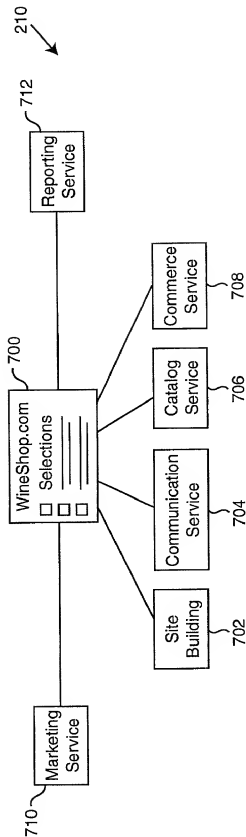


FIG. 7

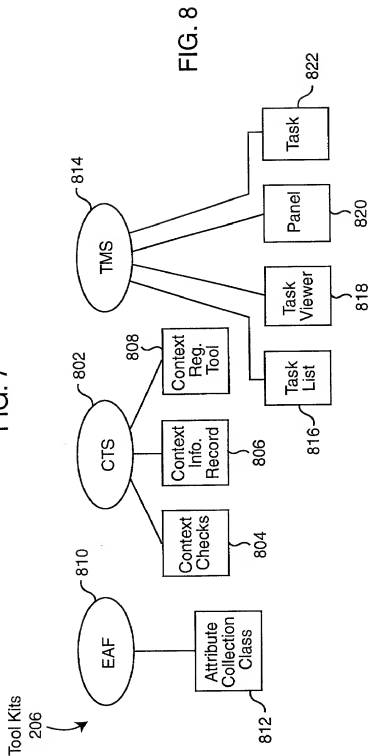


FIG. 8

FIG. 9A

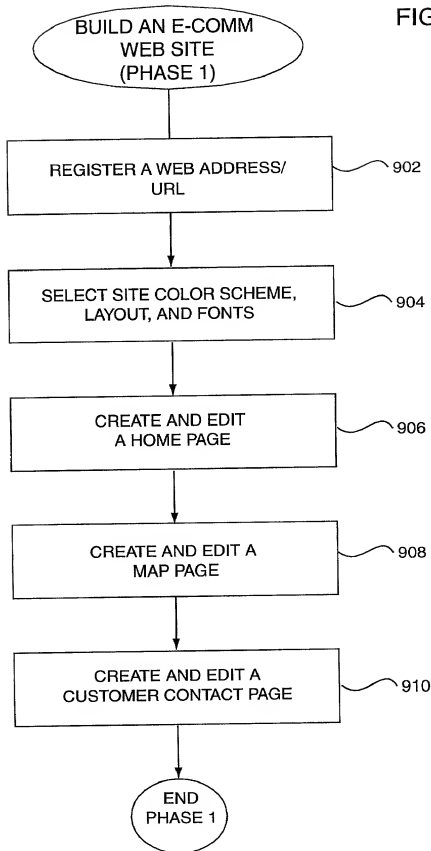


FIG. 9B

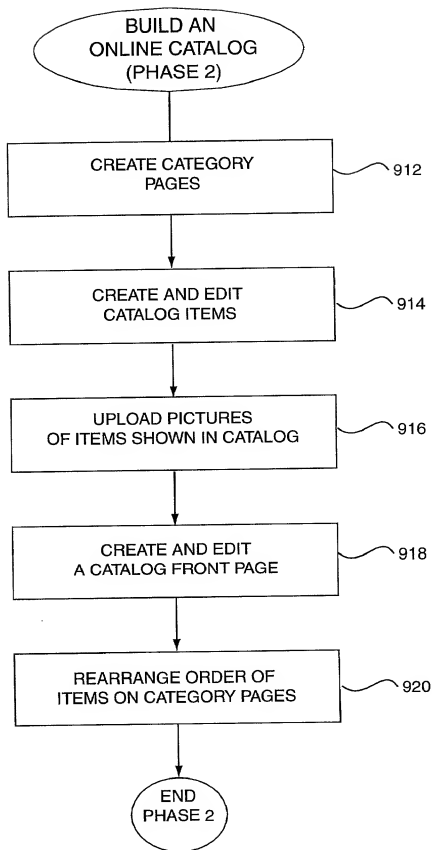


FIG. 9C

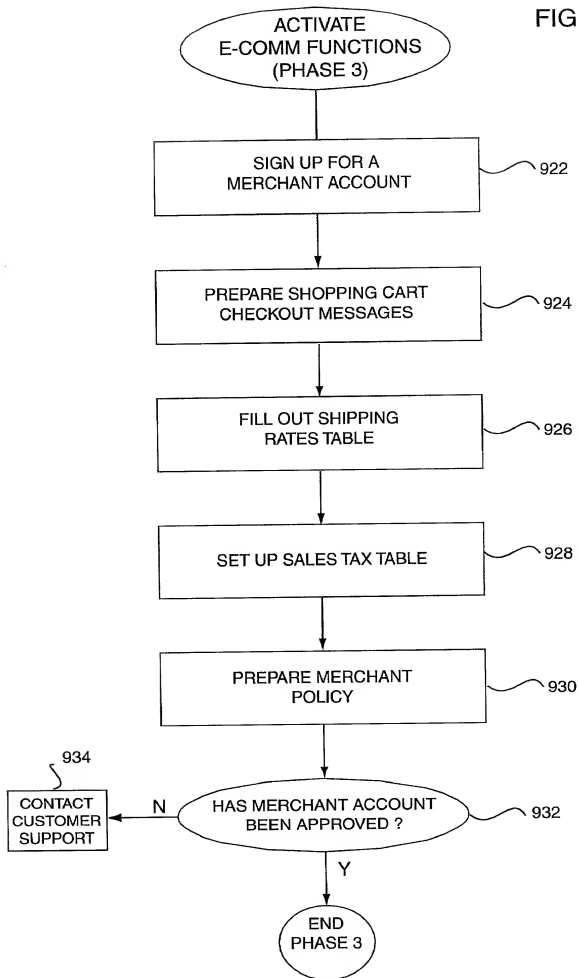
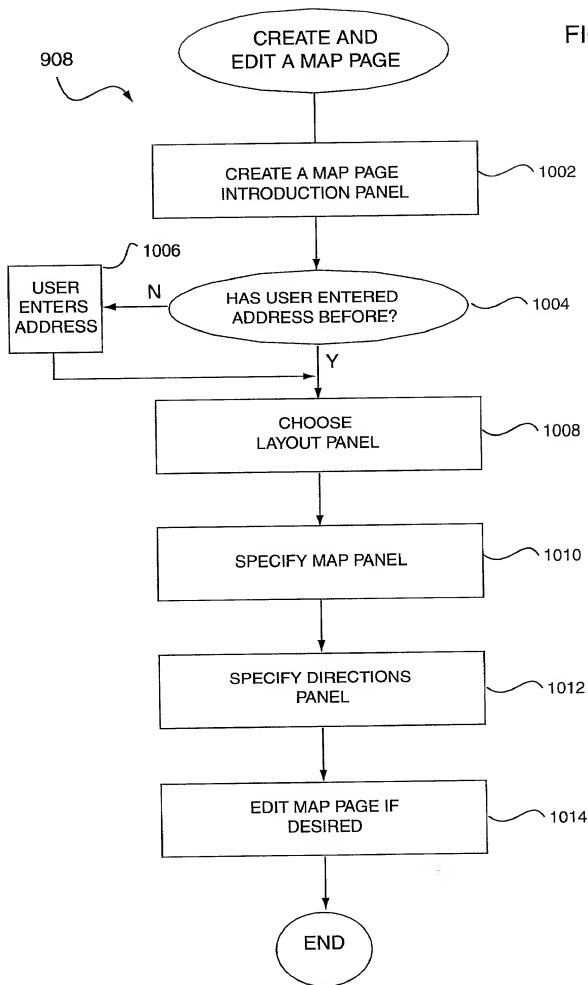


FIG. 10



002230 54560360

F10.11A

Bigstep.com

Site Building : Map and directions : **Pick a layout**

Your account | Help | Feedback | Sign out

Map and directions

Pick a layout

The pictures below show possible layouts for your map page. The gray and black lines in the pictures show you where your words will appear on the finished page.

Pick a layout.

1104

Address to right

Your address appears at the top right of the page above your map, separated by a thin line. Your directions' headline and text appear to the left of your map below the line. Your headline appears in bold.

1106

Address between lines

Your address appears at the top of the page, between two thin lines. A bold headline appears under your address, followed by your directions and other information.

1108

Address atop bar

Your address appears at the top of the page, placed over a thick bar of color. Your directions' headline appears below in bold, followed by your directions and other information.

FIG. 11B

Bigstep.com

Site Building : Map and directions : Create your map's appearance

Create your map's appearance

Here you'll decide how your map will be displayed the first time your customers view it. Your customers will zoom in and recenter the map interactively after they see the view you create here.

If you'd like to change the map's initial appearance you may recenter it, zoom in, or zoom out. Your change when you publish this page.

2601 Mission Street, 5th Floor, San Francisco, CA, 94110, US

NW

N

NE



© 1999 MapQuest.com, Inc. © 1999 Navigation Technologies

Use Subject to License / Copyright

SW

S

SE

To reorient the map, choose a direction.

For example, North (N), North East (NE), South (S), and so on. Click once on the letter next to the map, and your map will recenter in the direction you chose.

Zooming

To zoom your map in and out, change the level on the slider beside the map, or click the + buttons, above.

Choose Continue to go to the next step.

FIG. 11C

Bigstep.com

Site Building : Map and directions : Directions

Directions

If you'd like to add directions to your map page, do so here.



© 1999 MapQuest.com, Inc.; © 1999 Navigation Technologies

Write a headline for your directions.

For example, "Directions From Highway 101 South."

Add your directions here.

For example, "Take Highway 101 South to the Alemany Boulevard exit. Turn Right on Bayshore. Drive two

If you'd like to add more information to your map page, do so here.

Choose Continue to go to the next step.

**DECLARATION AND POWER OF ATTORNEY
FOR ORIGINAL U.S. PATENT APPLICATION**

EXPRESS MAIL NO.
EL560312993US

Attorney's Docket No. BIG1P001

As a below-named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe that I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled:

**AN INTEGRATED PLATFORM FOR DEVELOPING AND MAINTAINING A DISTRIBUTED
MULTIAPPLICATION ONLINE PRESENCE** the specification of which,

(check one)

1. ☒ is attached hereto.
2. ☐ was filed on _____ as
U.S. Application No. _____
and was amended on _____.
3. ☐ was filed on _____ as
International PCT Application No. _____
and was amended on _____.

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, CFR § 1.56.

Prior Foreign Application(s)

I hereby claim foreign priority benefits under Title 35, United States code, § 119(a)-(d) or § 365(b) of any foreign application(s) for patent or inventor's certificate, or § 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have identified below, by checking the box, any foreign application for patent or inventor's certificate, or PCT International application having a filing date before that of the application on which priority is claimed:

_____	_____	_____	Priority Benefits Claimed?
(Application No.)	(Country)	(Filing Date)	Yes ___ No ___
_____	_____	_____	Yes ___ No ___
(Application No.)	(Country)	(Filing Date)	

Provisional Application(s)

I hereby claim the benefit under 35 U.S.C. §119(e) of any United States provisional application(s) listed below:

<u>60/142,181</u>	<u>July 1, 1999</u>
(Application No.)	(Filing Date)
_____	_____
(Application No.)	(Filing Date)

Prior U.S. Application(s)

I hereby claim the benefit under Title 35, United States Code, § 120 of any United States application(s), or § 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of Title 35, United States Code, § 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, § 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application:

_____ (Application No.)	_____ (Filing Date)	_____ (Status - patented, pending, abandoned)
----------------------------	------------------------	--

_____ (Application No.)	_____ (Filing Date)	_____ (Status - patented, pending, abandoned)
----------------------------	------------------------	--

Power of Attorney

And I hereby appoint the law firm of **Beyer Weaver & Thomas, LLP** and all practitioners who are associated with the Customer Number 022434 as my principal attorneys to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith.

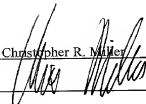
Direct Correspondence To:

Customer Number: 022434
BEYER WEAVER & THOMAS, LLP
P.O. Box 130
Mountain View, CA 94042-0130

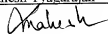
**Direct Telephone Calls To:****Rupak Nag at telephone number (510) 843-6200**

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Typewritten Full Name of**Sole or First Inventor:** Joshua A. Jacobs**Citizenship:** United States**Inventor's signature:** **Date of Signature:** 6/21/2000**Residence:** (City) San Francisco**(State/Country)** California/USA**Post Office Address:** 175 Bluxome, #124, San Francisco, CA 94107**Full Name of****Second Inventor:** John F. Shipley**Citizenship:** United States**Inventor's signature:** **Date of Signature:** 6/22/2000**Residence:** (City) San Francisco**(State/Country)** California/USA**Post Office Address:** 1303 Rhode Island, San Francisco, CA 94107-3246

Full Name of
Third Inventor: Christopher R. Miller
Inventor's signature: 
Residence: (City) Oakland
Post Office Address: 4025 Webster Avenue, Oakland, CA 94609

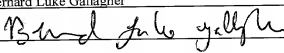
Citizenship: United States
Date of
Signature: 22 June 2000
(State/Country) California/USA

Full Name of
Fourth Inventor: Maresh Tvagarajan
Inventor's signature: 
Residence: (City) San Jose
Post Office Address: 3617 Copperfield Drive, #244, San Jose, CA 95136

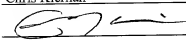
Citizenship: India
Date of
Signature: June 22, 2000
(State/Country) California/USA

Full Name of
Fifth Inventor: Kirt David Johnson
Inventor's signature: 
Residence: (City) Pacifica
Post Office Address: 1359 De Soto Drive, Pacifica, CA 94044

Citizenship: United States
Date of
Signature: 6/22/00
(State/Country) California/USA

Full Name of
Sixth Inventor: Bernard Luke Gallagher
Inventor's signature: 
Residence: (City) San Francisco
Post Office Address: 136 27th Street, San Francisco, CA 94110

Citizenship: United States
Date of
Signature: 6/22/00
(State/Country) California/USA

Full Name of
Seventh Inventor: Chris Kiernan
Inventor's signature: 
Residence: (City) San Francisco
Post Office Address: 233 Gough Street, San Francisco, CA 94102

Citizenship: United States
Date of
Signature: 6/22/00
(State/Country) California/USA